On the determination of probability density functions by using Neural Networks

Lluís Garrido[1,2], Aurelio Juste[2]

1) Dept. d'Estructura i Constituents de la Matèria,
Facultat de Física, Universitat de Barcelona,
Diagonal 647, E-08028 Barcelona, Spain.
Phone: +34 93 402 11 91   Fax: +34 93 402 11 98
e-mail: garrido@ecm.ub.es

2) Institut de Física d'Altes Energies,
Universitat Autònoma de Barcelona,
E-08193 Bellaterra (Barcelona), Spain.
Phone: +34 93 581 28 34   Fax: +34 93 581 19 38
e-mail: juste@ifae.es

Abstract

It is well known that the output of a Neural Network trained to disentangle between two classes has a probabilistic interpretation in terms of the a-posteriori Bayesian probability, provided that a unary representation is taken for the output patterns. This fact is used to make Neural Networks approximate probability density functions from examples in an unbinned way, giving a better performace than "standard binned procedures". In addition, the mapped p.d.f. has an analytical expression.

PACS'96: 02.50.Ph, 07.05.Kf, 07.05.Mh

(Submitted to *Comput. Phys. Commun.*)

1 Introduction

Estimating a probability density function (p.d.f.) in a n-dimensional space is a necessity which one may easily encounter in Physics and other fields. The standard procedure is to bin the space and approximate the p.d.f. by the ratio between the number of events falling inside each bin over the total and normalised to the bin volume. The fact of binning not only leads to a loss of information (which might be important unless the function is smoothly varying inside each bin) but is intrinsically arbitrary: no strong arguments for a defined binning strategy, e.g. constant bin size versus constant density per bin, exists. More sophisticated approaches imply for instance the definition of an “intelligent” binning, with smaller bins in the regions of rapid function variation. However, the main drawback still remains: even for a low number of bins per dimension, large amounts of data are necessary since the number of data points needed to fill the bins with enough statistical significance grows exponentially with the number of variables. As it will be shown, Neural Networks (NN) turn out to be useful tools for building up analytical n-dimensional probability density functions in an unbinned way from examples.

This manuscript is organised as follows: in Sect. 2 the proposed method to construct unbinned p.d.f.s from examples is described. After a brief introduction to the statistical interpretation of the output of a Neural Network applied to pattern recognition in the case of only two classes, an expression for the mapped p.d.f. is obtained. Then, a method to quantify the goodness of the mapped p.d.f. is described. In order to illustrate the concept, an artificial example is discussed in Sect. 3, whereas Sect. 4 is devoted to the discussion of an example of practical application in High Energy Physics. Finally, in Sect. 5, the conclusions are given.

2 Method

Let us assume that we have a sample of N events distributed among 2 different classes of patterns ($\mathcal{C}_1$ and $\mathcal{C}_2$), each event e being characterised by a set of n variables $\boldsymbol{x}^{(e)}$. Each class of patterns has a proportion α_i and is generated by the normalised probability density function $P_i(\boldsymbol{x})$, $i = 1, 2$ (in probability terms, $P_i(\boldsymbol{x}) = P(\boldsymbol{x} \mid \mathcal{C}_i)$ and $\alpha_i = P(\mathcal{C}_i)$).

By minimising over this sample the quadratic output-error E:

$$E\,[o] = \frac{1}{2N} \sum_{e=1}^{N} \left[o(\boldsymbol{x}^{(e)}) - d(\boldsymbol{x}^{(e)}) \right]^2 . \qquad (2.1)$$

with respect to the unconstrained function $o(\boldsymbol{x})$, where $d(\boldsymbol{x})$ takes the value 1 for the events belonging to class $\mathcal{C}_1$ and 0 for the events belonging to class $\mathcal{C}_2$, it can be shown [3, 4, 5, 6] that the minimum is achieved when $o(\boldsymbol{x})$ is the a-posteriori Bayesian probability to belong to class $\mathcal{C}_1$:

$$o^{(min)}(\boldsymbol{x}) = \mathcal{P}(\mathcal{C}_1 \mid \boldsymbol{x}). \qquad (2.2)$$

The above procedure is usually done by using layered feed-forward Neural Networks (see e.g. [1, 2] for an introduction). In this paper we have considered Neural Networks with topologies $N_i \times N_{h_1} \times N_{h_2} \times N_o$, where N_i ($N_o = 1$) are the number of input (ouput) neurons and N_{h_1}, N_{h_2} are the number of neurons in two hidden layers.

The input of neuron i in layer ℓ is given by,

$$I_i^\ell = \left\{ \begin{array}{cc} x_i^{(e)} & \ell = 1 \\ \sum_j w_{ij}^\ell S_j^{\ell-1} + B_i^\ell & \ell = 2,3,4 \end{array} \right. \tag{2.3}$$

where $x_i^{(e)}$ is the set of n variables describing a physical event e, the sum is extended over the neurons of the preceding layer $(\ell - 1)$, $S_j^{\ell-1}$ is the state of neuron j at layer $(\ell - 1)$ and B_i^ℓ is a bias input to neuron i at layer ℓ. The state of a neuron is a function of its input $S_j^\ell = F(I_j^\ell)$, where F is the neuron response function. In general the "sigmoid function", $F(I_j^\ell) = 1/(1 + e^{-I_j^\ell})$, is chosen since it offers a more sensitive modeling of real data than a linear one, being able to handle existing non-linear correlations. However, depending on the particular problem faced, a different neuron response function may be more convenient. For instance, in the artificial example described below, a sinusoidal neuron response function, $F(I_j^\ell) = (1 + \sin(I_j^\ell))/2$, has been adopted.

Back-propagation [7, 8, 9] is used as the learning algorithm. Its main objective is to minimise the above quadratic output-error E by adjusting the w_{ij} and B_i parameters.

Let us now consider the situation we are concerned in this paper: we have a large amount of events ("data") distributed according to the p.d.f. $\mathcal{P}_{data}(\boldsymbol{x})$, whose analytical expression is unknown and which we want precisely to approximate. If a Neural Network is trained to disentangle between those events and other ones generated according to any kwown p.d.f., $\mathcal{P}_{ref}(\boldsymbol{x})$ (not vanishing in a region where $\mathcal{P}_{data}(\boldsymbol{x})$ is non-zero), the Neural Network output will approximate, after training, the conditional probability for a given event to be of the "data" type:

$$o^{(min)}(\boldsymbol{x}) \simeq \mathcal{P}(data \mid \boldsymbol{x}) \equiv \frac{\alpha_{data}\mathcal{P}_{data}(\boldsymbol{x})}{\alpha_{data}\mathcal{P}_{data}(\boldsymbol{x}) + \alpha_{ref}\mathcal{P}_{ref}(\boldsymbol{x})}, \tag{2.4}$$

where α_{data} and α_{ref} are the proportions of each class of events used for training, satisfying $\alpha_{data} + \alpha_{ref} = 1$.

From the above expression it is straightforward to extract the NN approximation to $\mathcal{P}_{data}(\boldsymbol{x})$ as given by:

$$\mathcal{P}_{data}^{(NN)}(\boldsymbol{x}) = \mathcal{P}_{ref}(\boldsymbol{x}) \frac{\alpha_{ref}}{\alpha_{data}} \frac{o^{(min)}(\boldsymbol{x})}{1 - o^{(min)}(\boldsymbol{x})}. \tag{2.5}$$

As a result, the desired p.d.f. is determined in an unbinned way from examples. In addition, $\mathcal{P}_{data}^{(NN)}(\boldsymbol{x})$ has an analytical expression since we indeed have it for $\mathcal{P}_{ref}(\boldsymbol{x})$ and $o^{(min)}(\boldsymbol{x})$ is known once we have determined the network parameters (weights and bias inputs).

For what the reference p.d.f. is concerned, a good choice would be a p.d.f. built from the product of normalised good approximations to each 1-dimensional projection of the data p.d.f., thus making easier the learning of the existing correlations in the n-dimensional space. Since $\mathcal{P}_{ref}(\boldsymbol{x})$ is a normalised p.d.f. by construction, the normalisation of $\mathcal{P}^{(NN)}_{data}(\boldsymbol{x})$ will depend on the goodness of the Neural Network approximation to the conditional probability, so that in general it must be normalised a-posteriori. In the artificial (High Energy Physics) example shown below, the normalisation of the obtained p.d.f.s was consistent with 1 at the 1% (3%) level.

On the other hand, one would like to test the goodness of the approximation of the mapped p.d.f. to the true one. Given a data sample containing N_{data} events, it is possible to perform a test of the hypothesis of the data sample under consideration being consistent with coming from the mapped p.d.f. For that, one can compute the distribution of some test statistics like the log-likelihood function of Eq.(2.6), which can be obtained by generating Monte Carlo samples containing N_{data} events generated using the mapped p.d.f.

$$\mathcal{L} = \log(L) = \sum_{e=1}^{N_{data}} \log(\mathcal{P}^{(NN)}_{data}(\boldsymbol{x}^{(e)})) \tag{2.6}$$

Being $\mathcal{L}_{data}$ the value of the log-likelihood for the original data sample, the confidence level (CL) associated to the hypothesis of the data sample coming from the mapped p.d.f. is given by:

$$CL = \int_{-\infty}^{\mathcal{L}_{data}} d\mathcal{L}\, \mathcal{P}(\mathcal{L}) \tag{2.7}$$

which in practice can be obtained as the fraction of generated Monte Carlo samples of the data size having a value of the log-likelihood equal or below the one for the data sample. If the mapped p.d.f. is a good approximation to $\mathcal{P}_{data}$, the expected distribution for CL evaluated for different data samples should have a flat distribution as it corresponds to a cumulative distribution.

3 Artificial example

In this section we propose a purely artificial example in order to illustrate how a Neural Network can perform a mapping of a 5-dimensional p.d.f. in an unbinned way from examples.

In this example our "data" will consist in a sample of 100000 events generated in the cube $[0,\pi]^5 \in \mathbf{R}^5$ according to the following p.d.f.:

$$\mathcal{P}_{data}(\boldsymbol{x}) = \frac{1}{C}\left(\sin(x_1+x_2+x_3)+1\right)\left(\frac{\sin(x_4^2+x_5^2)}{x_4^2+x_5^2}+1\right), \tag{3.1}$$

which we want to estimate from the generated events. In the above expression, C is a normalisation factor such that $\mathcal{P}_{data}(\boldsymbol{x})$ has unit integral. The above p.d.f. has a rather intrincate structure of maxima and minima in both, the

3-dimensional space of the first three variables and the 2-dimensional space of the two last variables.

In order to map the above p.d.f., we need to train a Neural Network to disentangle between events generated according to $\mathcal{P}_{data}(\boldsymbol{x})$ and events generated according to any $\mathcal{P}_{ref}(\boldsymbol{x})$ non-vanishing in any region where $\mathcal{P}_{data}(\boldsymbol{x})$ is different from zero. In order to make easier the learning of the existing correlations in the 5-dimensional space, as explained before, $\mathcal{P}_{ref}(\boldsymbol{x})$ is chosen as the product of good approximations to the 1-dimensional projections of $\mathcal{P}_{data}(\boldsymbol{x})$, properly normalised to have unit integral.

In the case of data p.d.f., it turns out that the 1-dimensional projections of the three first variables are equal and essentially flat, whereas the 1-dimensional projections for the two last variables can be parametrised as a 4th degree polinomial (P_4). Therefore, we choose as reference p.d.f.:

$$\mathcal{P}_{ref}(\boldsymbol{x}) = \frac{1}{C'}\, P_4(x_4) \cdot P_4(x_5) \tag{3.2}$$

and generate a number of 100000 events according to it. As before, C' is a normalisation factor so that $\mathcal{P}_{ref}(\boldsymbol{x})$ has unit integral.

After the training and normalisation, the p.d.f. given by Eq.(2.5) constitutes a reasonably good approximation to $\mathcal{P}_{data}(\boldsymbol{x})$, as it is indeed observed in Fig. 1, where both are compared for different slices in the 5-dimensional space with respect to the variable x_1. For comparison, it is also shown the reference p.d.f. which, as expected, is unable to reproduce the complicated structure of maxima and minima in the 5-dimensional space.

As explained in previous section, it is posible to perform a test of the goodness of the mapped p.d.f. For that, a number of 10000 Monte Carlo samples have been generated with the mapped p.d.f., each one containing 100000 events, which is the same number of events of the "data" sample. The log-likelihood is computed for each MC sample and its distribution is shown in Fig. 2a), in which the arrow indicates the value of the log-likelihood for the original data sample ($\mathcal{L}_{data}$). From this distribution and the value of $\mathcal{L}_{data}$ we have found a confidence level of 5.5% associated to the hypothesis of the data sample coming from the mapped p.d.f. This seems a low CL and needs further comments, but as we know the true p.d.f given by Eq.(2.5), we can do much better than performing a single measurement for CL and is to find out its distribution.

Very often in High Energy Physics and other fields the problem consist on estimating a p.d.f. from a sample of simulated Monte Carlo events which is much larger (typically a factor 100 times larger) than the experimental data sample over which we should use this p.d.f (see the High Energy Physics example of Sect. 4). For this reason we have obtained the CL distribution in three different scenarios: when the number of experimental data events (N_{exp}) has the same number of events as the data sample used to obtain the mapped p.d.f. ($N_{data} = 100000$), and two with smaller statistics, one with $N_{exp} = 10000$ and another with $N_{exp} = 1000$.

A number of 10000 Monte Carlo samples have been generated with the mapped p.d.f., each containing N_{exp} events, for the three different values of N_{exp} and the log-likelihood is computed for each sample in all three scenarios.

On the other hand, a number of 1000 data samples are generated with the true p.d.f. in the three scenarios and the confidence level is computed according to Eq.(2.7). The distribution of CL is shown in Fig. 2b) for $N_{exp} = 1000$ (dotted line), 10000 (dashed line) and 100000 (solid line). It can be observed that for $N_{exp} = 1000$ the distribution of CL is to a good approximation a flat distribution whereas for $N_{exp} = 10000$ it starts deviating from being flat, which indicates that the statistics of the data sample is high enough to start "detecting" systematic deviations in the mapped p.d.f. with respect to the true one.

In the case of $N_{exp} = 1000$ which, as mentioned above illustrates a common situation in High Energy Physics, the mapped p.d.f. turns out to be a good enough approximation when used for the smaller experimental data sample. In the other extreme, $N_{exp} = 100000$, which illustrates the situation in which there is a unique data sample from which one wants to estimate the underlying p.d.f., it can be observed in Fig. 2b) (solid line) the existence of enough resolution to detect systematic deviations in the mapped p.d.f. with respect to the true one. It should be stressed the very complicated structure of the true p.d.f., which makes extremely difficult its accurate mapping and nevertheless the difference between both distributions are the ones observed in Fig. 1 between the solid and the dashed lines. In such situations we can not use the mapped p.d.f. for fine probability studies but it is clear that it is still very useful for other kind of studies like classification or discrimination.

4 High Energy Physics example

In order to illustrate the practical interest of p.d.f. mapping, the following High Energy Physics example is considered.

One of the major goals of LEP200 is the precise measurement of the mass of the W boson. At energies above the WW production threshold ($\sqrt{s} > 161$ GeV) W bosons are produced in pairs and with sufficient boost to allow a competitive measurement of the W mass by direct reconstruction of its product decays. Almost half of the times (45.6%) both W bosons decay hadronically, so that four jets of particles are observed in the final state.

Most of the information about the W mass is contained in the reconstructed di-jet invariant mass distribution, so that M_W can be estimated by performing a likelihood fit to this 2-dimensional distribution. Therefore, the W mass estimator, $\hat{M_W}$, is obtained by maximising the log-likelihood function:

$$\mathcal{L}(M_W) = \sum_{e=1}^{N} \log \mathcal{P}(s_1^{\prime(e)}, s_2^{\prime(e)} \mid M_W) \tag{4.1}$$

with respect to M_W, where $\mathcal{P}(s_1^{\prime(e)}, s_2^{\prime(e)} \mid M_W)$ represents the probability of event e, characterised by the two measured invariant masses $(s_1^{\prime(e)}, s_2^{\prime(e)})$, given M_W which, accounting for the existing background, can be expressed as:

$$\mathcal{P}(s_1', s_2' \mid M_W) = \rho_{ww}\mathcal{P}_{ww}(s_1', s_2' \mid M_W) + (1 - \rho_{ww})\mathcal{P}_{bckg}(s_1', s_2'). \tag{4.2}$$

In the above expression ρ_{ww} is the expected signal purity in the sample and $\mathcal{P}_{ww}$ and $\mathcal{P}_{bckg}$ are respectively the p.d.f. for signal (W-pair production) and background in terms of the reconstructed di-jet invariant masses. For a typical selection procedure above threshold at LEP200, signal efficiencies in excess of 80% with a purity at the level 80% can be obtained in the fully hadronic decay channel.

Therefore, in order to determine M_W, we need to obtain both p.d.f.s, for signal and background, in terms of the reconstructed di-jet invariant masses.

At $\sqrt{s} = 172$ GeV and after selection, most of the background comes from QCD. To map the p.d.f. for the background, a 2-5-2-1 Neural Network was trained with ~ 6000 selected $q\bar{q}$ Monte Carlo events generated with full detector simulation ("data") and the same number of "reference" Monte Carlo events generated according to the 1-dimensional projections of the "data" sample.

As far as the signal p.d.f. is concerned, it depends on the parameter we want to estimate: M_W. It can be obtained by a folding procedure of the theoretical prediction for the 3-fold differential cross-section in terms of the 2 di-quark invariant masses (s_1 and s_2) and x (the fraction of energy radiated in the form of initial state photons), with a transfer function T, which accounts for distortions in the kinematics of the signal events due to fragmentation, detector resolution effects and biases in the reconstruction procedure. This transfer function represents the conditional probability of the reconstructed invariant masses given some invariant masses at the parton level and initial state radiation (ISR). The ISR is most of the times lost along the beam pipe and therefore unknown, reason for which it must be integrated over. This conditional probability is given by:

$$T(s_1', s_2' \mid s_1, s_2, x) = \frac{f(s_1', s_2', s_1, s_2, x)}{g(s_1, s_2, x)}, \tag{4.3}$$

where s_i' stands for each reconstructed invariant mass and $g(s_1, s_2, x)$ is theoretically known and has a compact expression, reason for which there is no need to map it.

Then, the goal is to map the 5-dimensional p.d.f. $f(s_1', s_2', s_1, s_2, x)$. To do it, a 5-11-5-1 Neural Network was trained with 40000 hadronic WW Monte Carlo events generated with full detector simulation ("data") and the same number of "reference" events generated according to the 1-dimensional projections of the "data" sample.

In order to test that the event-by-event p.d.f. is meaningful, the predicted 1-dimensional projection of the average invariant mass distribution is compared to Monte Carlo in Figs. 3a) and b) for both signal and background by using the obtained $\mathcal{P}_{ww}$ and $\mathcal{P}_{bckg}$, respectively. Note the overall good agreement between the distributions.

The unbiasedness of the obtained estimator is checked by computing the calibration curve with respect the true parameter by performing a large number of fits to Monte Carlo samples generated with different values of M_W.

The performance of the NN in mapping a n-dimensional p.d.f. has been compared to the "box method" [10], a standard procedure to build up binned p.d.f.s. In the case of the background p.d.f., which is only 2-dimensional, the "box method" yielded reasonable results as shown in Fig. 3b), while in the case of the 5-dimensional p.d.f. it showed strong limitations which made impossible its application. The main reason is the time required to compute the final p.d.f which needs an integration on top of the adjustement of the "box method" parameters (initial box size, minimum number of MC points inside each box, etc) in a space of high dimensionality and limited statistics. Is in this environment where the mapping of p.d.f.s by means of NNs may be superior to "standard binned procedures" in terms of accuracy (the p.d.f. is determined in an unbinned way from examples) and speed (the resulting p.d.f. is an analytic function).

5 Conclusions

We have shown that Neural Networks are useful tools for building up n-dimensional p.d.f.s from examples in an unbinned way. The method takes advantage of the interpretation of the Neural Network output, after training, in terms of a-posteriori Bayesian probability when a unary representation is taken for the output patterns. A purely artificial example and an example from High Energy Physics, in which the mapped p.d.f.s are used to determine a parameter through a maximum likelihood fit, have also been discussed. In a situation of high dimensionality of the space to be mapped and limited available statistics, the method is superior to "standard binned procedures".

6 Acknowledgements

This research has been partly supported by CICYT under contract number AEN97-1697.

References

[1] J.A. Hertz, A. Krogh and R.G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley, Redwood City, California (1991).

[2] B. Müller and J. Reinhardt, *Neural networks: an introduction*, Springer-Verlag, Berlin (1991).

[3] Ll. Garrido and S. Gómez, Analytical interpretation of feed-forward nets outputs after training, *Int. J. of Neural Systems* **7** (1996) 19.

[4] A. Papoulis, *Probability, random variables and stochastic processes*, McGraw-Hill, New York (1965).

[5] D.W. Ruck, S.K. Rogers, M. Kabriski, M.E. Oxley and B.W. Suter, The multilayer perceptron as an approximation to a Bayes optimal discriminant function, *IEEE Trans. Neural Networks* **1** (1990) 296.

[6] E.A. Wan, Neural network classification: a Bayesian interpretation, *IEEE Trans. Neural Networks* **1** (1990) 303.

[7] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, *Nature* **323** (1986) 533.

[8] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propagation. In *Parallel Distributed Processing*, D.E. Rumelhart and J.L. McClelland (eds.), MIT Press, Vol. 1, Cambridge, MA (1986) 318.

[9] P. Werbos, *Beyond regression: new tools for prediction and analysis in the behavioral sciences*, Ph.D. thesis, Harvard University (1974).

[10] D.M. Schmidt, R.J. Morrison and M.S. Witherell, A general method of estimating physical parameters from a distribution with acceptance and smearing effects, *Nucl. Inst. and Meth.* **A328** (1993) 547.

Figure captions

- **Figure 1:** Comparison between the true (solid line) and the mapped (dashed line) and the reference (dotted line) p.d.f. versus x_1 for different slices in the 5-dimensional space: (a) $x_2 = x_3 = x_4 = x_5 = 0$, (b) $x_2 = x_1$, $x_3 = x_4 = x_5 = 0$, (c) $x_3 = x_2 = x_1$, $x_4 = x_5 = 0$ and (d) $x_4 = x_3 = x_2 = x_1$, $x_5 = 0$.

- **Figure 2:** (a) Distribution of the log-likelihood computed for Monte Carlo samples of 100000 events generated according to the mapped p.d.f. The arrow indicates the value of the log-likelihood for the original data sample. (b) Distribution of the confidence level for data samples containing 1000 (dotted line), 10000 (dashed line) and 100000 (solid line) events respectively, generated with the true p.d.f., of being consistent with the hypothesis of coming from the mapped p.d.f.

- **Figure 3:** Comparison between NN (solid line) and Monte Carlo (points with error bars) prediction for the average di-jet invariant mass p.d.f. for a) signal and b) background. In b), the p.d.f. as obtained by a box method (dashed line) is also shown.

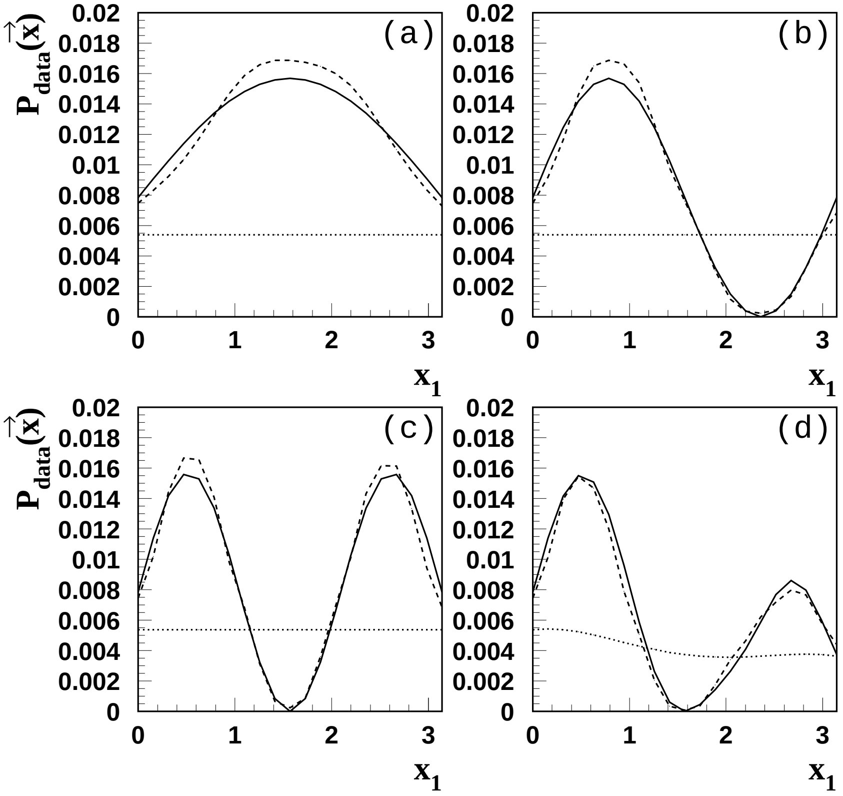

Figure 1: Comparison between the true (solid line), the mapped (dashed line) and the reference (dotted line) p.d.f. versus x_1 for different slices in the 5-dimensional space: (a) $x_2 = x_3 = x_4 = x_5 = 0$, (b) $x_2 = x_1$, $x_3 = x_4 = x_5 = 0$, (c) $x_3 = x_2 = x_1$, $x_4 = x_5 = 0$ and (d) $x_4 = x_3 = x_2 = x_1$, $x_5 = 0$.

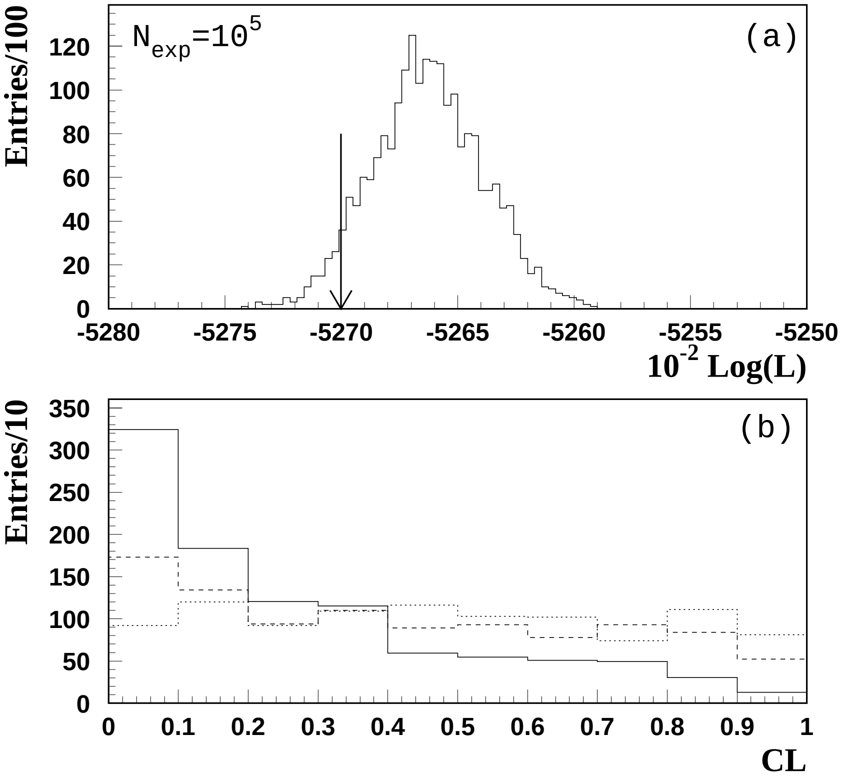

Figure 2: (a) Distribution of the log-likelihood computed for Monte Carlo samples of 100000 events generated according to the mapped p.d.f. The arrow indicates the value of the log-likelihood for the original data sample. (b) Distribution of the confidence level for data samples containing 1000 (dotted line), 10000 (dashed line) and 100000 (solid line) events respectively, generated with the true p.d.f., of being consistent with the hypothesis of coming from the mapped p.d.f.

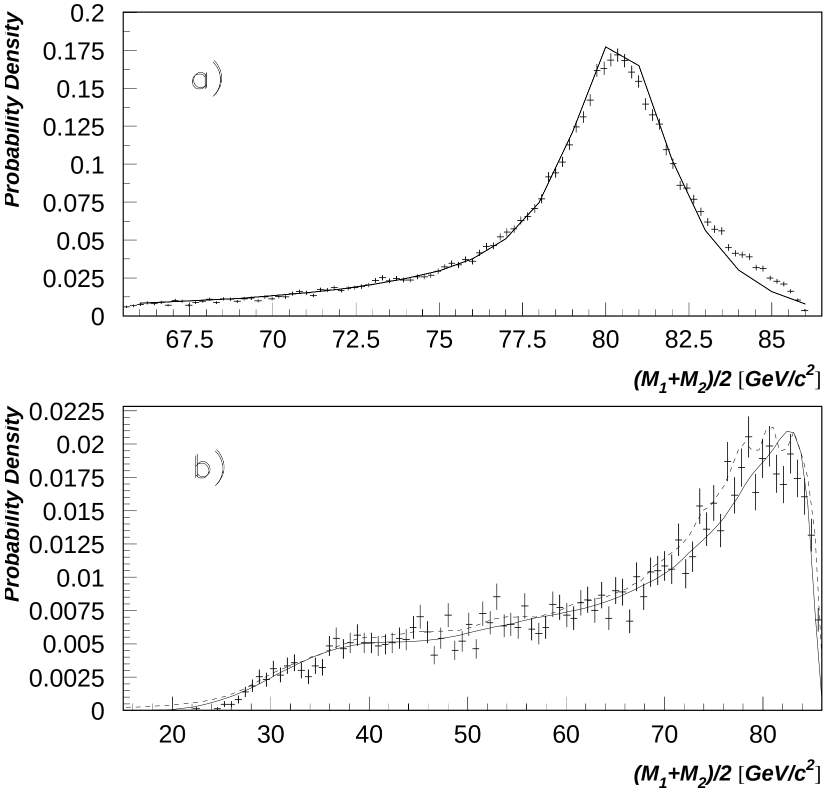

Figure 3: Comparison between NN (solid line) and Monte Carlo (points with error bars) prediction for the average di-jet invariant mass p.d.f. for a) signal and b) background. In b), the p.d.f. as obtained by a box method (dashed line) is also shown.